Applied Computational Economics - Winter 2014
ECON 647

Professor Barczyk

Office: LEA 321b
E-mail: daniel.barczyk@mcgill.ca
Office Phone: 514-398-4400 Ext. 09057

Class Location: LEA 517
Class Time: Tue & Thu 1:05pm - 2:25pm
Office Hours: by appointment

Textbook
We will closely follow the book *Applied Computational Economics and Finance* by Mario J. Miranda and Paul Fackler. A deeper and wider coverage of the numerical methods discussed may be found in *Numerical Methods in Economics* by Kenneth L. Judd. A good introduction to dynamic programming with numerical techniques is the book *Dynamic Economics: Quantitative Methods and Applications* by Jerome Adda and Russell W. Cooper.

Course description
This is a graduate course with the goal to familiarize you with some numerical methods that can be used to numerically obtain solutions to economic problems for which closed-form solutions are unavailable. The first part of the course focuses on basic numerical methods and the second part on solving dynamic models in economics and finance. Throughout the emphasis is on applications. In order to perform computations we will be using Matlab. I do not assume that you have any previous experience with Matlab or any other computing software. The prerequisites for the course are linear algebra and multivariable calculus.

Problem sets
For the problem sets you are encouraged to work in groups of two or three. Your write-up should be typed and include:

1. a brief summary of the methods used;
2. tables and/or graphs to describe your results;
3. your computer programs in an appendix.

Class presentations
You present the solutions to the problem sets in a structured, seminar-style way. You should discuss the theory and the solution method underlying the given problem and show how you have solved the problem using Matlab.

Term paper
You have considerable flexibility what you do for your term paper. You must however use some of the computational methods from the course. For example, you can replicate results from an existing paper, extend an existing paper, or ideally come up with your own research

question. In either case you will have to start early to think about what you want to do and how you are going to do it. Even replicating existing results can be time consuming as you have to find an appropriate paper (which is part of the challenge) that is feasible and interesting to you.
I strongly recommend you to take this opportunity seriously since this may help you in your current and/or future research. You have to send me a proposal the latest by **April 1**. The paper is due **May 1**. Please send me your paper and an accompanying code toolbox by email. The toolbox should be structured in such a way that the results are easily replicable, i.e. there should be a main program that calls all the other functions and returns the results in tables and/or graphs. Make sure to document the code along the way so that I can see the various steps and assumptions that you make (more importantly you will know later on what you have done). There is no need to have a long paper. Concentrate most of your efforts to find an interesting topic and on the computation. If you run into computational difficulties explore these: most likely you made a mistake but sometimes you might find something interesting or you discover that the method you have chosen is not suitable.

Grades
PhD students (no presentations)
Your grade is be based on the problem sets (50%) and the term paper (50%).

M.A. students (no term paper)
Your grade is based on the problem sets (50%), class presentations (40%), and class participation (10%).

Course Outline

Part I: Basic Numerical Methods

1. Linear equations: Ill conditioning, LU decomposition, and iterative methods.
2. Nonlinear equations: Nonlinear root-finding problems and nonlinear fixed point problems.
3. Complementarity problems: Compute economic models in which control variables are subject to constraints.
4. Numerical integration and differentiation. Solving ordinary differential equations.
5. Function approximation: The interpolation and collocation methods. Solving functional equations approximately.

Part II: Dynamic Programming

1. Discrete time, *discrete* state dynamic (Markov) models. Economic examples. Numerical solution algorithms: Backward recursion, function iteration, and policy iteration. Simulation and analysis.

2. Discrete time, *continuous* state dynamic (Markov) models. Theory and examples of dynamic optimization and dynamic equilibrium models such as American option pricing, economic growth, dynamic games, and asset pricing models.

3. Methods to solve discrete time, *continuous* state dynamic (Markov) models.

4. Depends on how much time is left...

Academic Integrity
McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offenses under the code of student conduct and disciplinary procedures (see http://www.mcgill.ca/integrity/ for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples renseignements, veuillez consulter le site http://www.mcgill.ca/integrity).